

---

# Quality surface meshing using discrete parametrizations

Emilie Marchandise<sup>1</sup>, Jean-Francois Remacle<sup>1</sup>, and Christophe Geuzaine<sup>2</sup>

<sup>1</sup> Université catholique de Louvain, Institute of Mechanics, Materials and Civil Engineering (iMMC), Place du Levant 1, 1348 Louvain-la-Neuve, Belgium  
`emilie.marchandise@uclouvain.be`, `jean-francois.remacle@uclouvain.be`

<sup>2</sup> Université de Liège, Department of Electrical Engineering and Computer Science, Montefiore Institute B28, Grande Traverse 10, 4000 Liège, Belgium  
`cgeuzaine@ulg.ac.be`

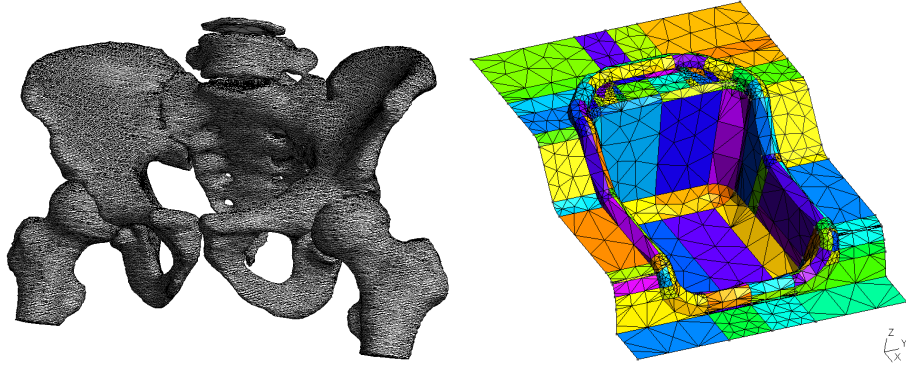
**Summary.** We present 3 mapping/flattening techniques for triangulations of poor quality triangles. The implementation of those mappings as well as the boundary conditions are presented in a very comprehensive manner such that it becomes accessible to a wider community than the one of computer graphics. The resulting parameterizations are used to generate new triangulations or quadrilateral meshes for the model that are of high quality.

## 1 Introduction

There are two kind of applications for which it might be desirable to remesh a 3D surface (see Fig. 1).

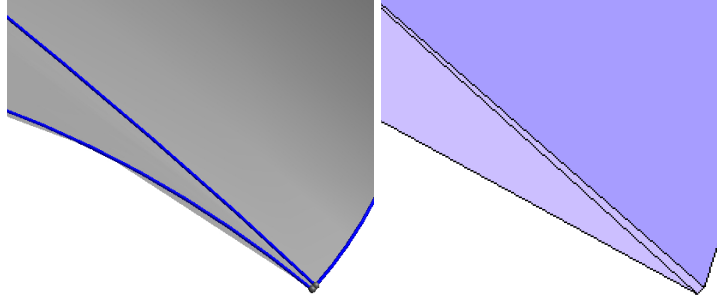
The first application concerns medical geometries that are often described only by a triangulation (in stereolithography STL format). This triangulation is the result of a segmentation procedure from the CT scan or MRI dicom-images. Those triangulations can be oversampled and have triangles of poor quality with small elementary angles. Those low quality meshes are not suitable for finite element simulations since the quality of the mesh will impact both on the accuracy and efficiency of the numerical method [35, 2]. In this case, it is desirable to build a high quality mesh from those low quality meshes before performing any numerical simulation.

The other application is about CAD models. CAD models are often made of a huge amount of patches that have no physical significance and a straightforward meshing of those patches often leads to meshes that are not suitable for finite element simulations. Indeed, as most surface mesh algorithms mesh model faces individually, mesh points are generated on the bounding edges of those patches and if thin CAD patches exist in the model they will result in the creation of small distorted triangles with very small angles (Fig.2).



**Fig. 1.** Examples of geometries for which a remeshing procedure is desirable. Left figure shows an example of an oversampled STL triangulation resulting from a mesh segmentation of a human pelvis and right figure shows the straightforward meshing of a CAD geometry of a maxi-cosi.

Those low quality elements present in the surface mesh will often hinder the convergence of the FE simulations on those surface meshes. Besides, they also prevent the generation of quality volumetric meshes for three-dimensional finite element computations (CFD, structure mechanics, etc.). An efficient manner to build a high quality mesh for those CAD models is then to build from the initial CAD mesh a cross-patch parametrization that enables the remeshing of merged patches.



**Fig. 2.** Example of 2 patches of a CAD geometry (left) for which the mesh (right) contains a very small triangle of poor quality.

There are mainly two approaches for surface remeshing: mesh adaptation strategies [18, 3, 38] and parametrization techniques [6, 40, 26, 36, 19, 22].

Mesh adaptation strategies use local mesh modifications in order both to improve the quality of the input surface mesh and to adapt the mesh to a given mesh size criterion. In parametrization techniques, the input mesh serves as a support for building a continuous parametrization of the surface. (In the case of CAD geometries, the initial mesh can be created using any off the shelf surface mesher for meshing the individual patches.) Surface parametrization techniques originate mainly from the computer graphics community: they have been used extensively for applying textures onto surfaces [5, 23] and have become a very useful and efficient tool for many mesh processing applications [9, 15, 21, 33, 12]. In the context of remeshing procedures, the initial surface is parametrized onto a surface in  $\mathcal{R}^2$ , the surface is meshed using any standard planar mesh generation procedure and the new triangulation is then mapped back to the original surface [8, 27].

The existing methods for discrete parametrization can be classified as follows: linear, non-linear and hybrid methods. Non-linear methods based on discrete or differential-geometric non-linear distortion measure [17, 34, 41] offer strong guarantees on the absence of triangle folding and flipping at the cost of a generally higher computational effort. Some authors have also suggested hybrid techniques that linearize those non-linear measures at the cost of only a few linear solves [4, 39]. Linear methods require only the resolution of a single linear system. Most methods require to map the vertex of the patch boundary to a given polygon (usually convex) in the parametric plane. This is for example the case of the discrete harmonic map introduced by Eck [8] or the more robust convex combination map of Floater [9]. Some authors also suggested extensions to free boundaries by pinning down only two vertices. This is the case for example in the least square conformal maps (LSCM) introduced by Levy et al. [21] and the discrete conformal parametrizations (DCP) of Desbrun et al. [1]. These mappings could achieve lower angle distortion than previous results. However, as the quality of the parametrization can depend significantly on the choice of the constrained vertices, Mullen et al. [28] suggested to spread the constraints throughout the mesh by constraining that the barycenter of the mapping must be at  $(0, 0)$  and that the moment of inertia of the boundary must be unit. In [28], those spread constraints are taken into account through recourse to spectral theory.

In this paper, we present and compare three different types of linear harmonic maps for the discrete parametrization of triangulated surfaces. The implementation of those mappings as well as the boundary conditions are presented in a very comprehensive manner such that it becomes accessible to a wider community than the one of computer graphics. The discrete parametrization aims at computing the discrete mapping  $\mathbf{u}(\mathbf{x})$  that maps every triangle of the three dimensional surface  $\mathcal{S}$  to another triangle of  $\mathcal{S}'$  that has a well known parametrization:

$$\mathbf{x} \in \mathcal{S}_{\mathcal{T}} \subset \mathcal{R}^3 \mapsto \mathbf{u}(\mathbf{x}) \in \mathcal{S}'_{\mathcal{T}} \subset \mathcal{R}^2 \quad (1)$$

We restrict ourselves to the parametrization of non-closed triangulated surfaces since we have already presented in previous papers [25, 24] efficient techniques to split a closed object into a series of different patches in the context of surface remeshing.

The overall procedure is implemented in the open-source mesh generator Gmsh [14].

## 2 Harmonic energy minimizations

A harmonic map minimizes distortion in the sense that it minimizes the Dirichlet energy of the mapping  $\mathbf{u}(\mathbf{x})$ :

$$E_D(\mathbf{u}) = \int_{\mathcal{M}} \frac{1}{2} |\nabla \mathbf{u}|^2 ds. \quad (2)$$

subject to Dirichlet boundary conditions  $\mathbf{u} = \mathbf{u}_D$  on  $\partial\mathcal{M}^i$ . Harmonic maps are not in general conformal and do not preserve angles but they are popular since they are very easy to compute and are guaranteed to be one-to-one for convex regions [29, 7].

## 3 Convex combination map

In contrast to the continuous harmonic map (2), Floater et al. showed in [10, 11] that the discrete version of (2) is not always one-to-one. To ensure a discrete maximum principle, the authors introduced a convex combination. One particular type of convex combination maps is for example the barycentric mapping by Tutte [37] that asks every interior vertex  $\mathbf{u}_i$  be the barycenter of its neighbors.

$$\mathbf{u}_i = \sum_{k=1}^{d_i} \lambda_k \mathbf{u}_j, \quad \sum_{k=1}^{d_i} \lambda_k = 1, \quad (3)$$

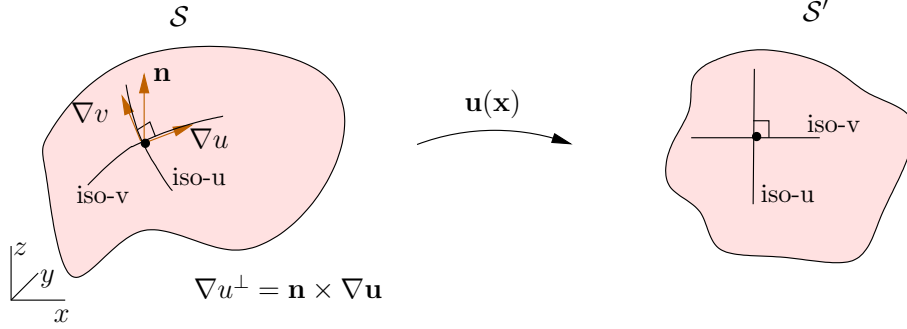
where  $d_i$  denotes the number of vertices that are neighbors to node  $i$ .

## 4 Least Square Conformal map

The least square conformal map as introduced by Levy et al. [21] asks that the gradient of  $u$  and the gradient of  $v$  be as orthogonal as possible in the parametrization and have the same norm. This can be seen as an approximation of the Cauchy-Riemann equations. For a piecewise linear mapping, the least square conformal map can be obtained by minimizing the energy:

$$E_{\text{LSCM}}(\mathbf{u}) = \int_{\mathcal{M}} \frac{1}{2} |\nabla u^\perp - \nabla v|^2 ds, \quad (4)$$

where  $^\perp$  denotes a counterclockwise  $90^\circ$  rotation in  $\mathcal{S}$ . For a 3D surface defined by a normal vector  $\mathbf{n}$ , the counterclockwise rotation of the gradient can be written as:  $\nabla u^\perp = \mathbf{n} \times \nabla \mathbf{u}$  (see Fig.3).



**Fig. 3.** Definitions for a conformal mapping.  $\nabla u^\perp$  denotes the counterclockwise  $90^\circ$  rotation of the gradient  $\nabla u$  for a 3D surface.

Equation (4) can be simplified and rewritten as follows:

$$\begin{aligned} E_{\text{LSCM}}(\mathbf{u}) &= \int_{\mathcal{M}} \frac{1}{2} (\nabla u^\perp \cdot \nabla u^\perp + \nabla v \cdot \nabla v - 2\nabla u^\perp \cdot \nabla v) ds \\ &= \int_{\mathcal{M}} \frac{1}{2} (\nabla u \cdot \nabla u + \nabla v \cdot \nabla v - 2(\mathbf{n} \times \nabla u) \cdot \nabla v) ds. \end{aligned} \quad (5)$$

Recalling the identity that a “dot” and a “cross” can be interchanged without changing the result, we have

$$E_{\text{LSCM}}(\mathbf{u}) = \int_{\mathcal{S}} \frac{1}{2} (\nabla u \cdot \nabla u + \nabla v \cdot \nabla v - 2\mathbf{n} \cdot (\nabla u \times \nabla v)) ds. \quad (6)$$

## 5 Discrete harmonic maps with finite elements

We now derive the finite element formulation of the quadratic minimisation problems (2)-(4). We denote by the functional  $J$  either the Dirichlet energy  $E_D$  or the least-square conformal energy  $E_{\text{LSCM}}$  to be minimized:

$$\min_{\mathbf{u} \in U(\mathcal{M})} J(\mathbf{u}), \quad \text{with} \quad \mathbf{U}(\mathcal{S}) = \{\mathbf{u} \in H^1(\mathcal{S}), \mathbf{u} = \mathbf{u}_D(\mathbf{x}) \text{ on } \partial\mathcal{M}^i\}. \quad (7)$$

We assume the following finite expansions for  $\mathbf{u}$

$$\mathbf{u}_h(\mathbf{x}) = \sum_{i \in I} \mathbf{u}_i \phi_i(\mathbf{x}) + \sum_{i \in J} \mathbf{u}_D(\mathbf{x}_i) \phi_i(\mathbf{x}) \quad (8)$$

where  $I$  denotes the set of nodes of  $\mathbf{M}$  that do not belong to the Dirichlet boundary, where  $J$  denotes the set of nodes of  $\mathbf{M}$  that belong to the Dirichlet boundary and where  $\phi_i$  are the nodal shape functions associated to the nodes of the mesh. We assume here that the nodal shape function  $\phi_i$  is equal to 1 on vertex  $\mathbf{x}_i$  and 0 on any other vertex:  $\phi_i(\mathbf{x}_j) = \delta_{ij}$ .

Thanks to expansion Eq. (8), the functional  $J$  defining the energy of the least square conformal map Eq. (6) can be rewritten as

$$\begin{aligned}
J(\mathbf{u}_1, \dots, \mathbf{u}_N) = & \frac{1}{2} \sum_{i \in I} \sum_{j \in I} u_i u_j \int_{\mathbf{M}} \nabla \phi_i \cdot \nabla \phi_j \, ds + \sum_{i \in I} \sum_{j \in J} u_i u_D(\mathbf{x}_j) \int_{\mathbf{M}} \nabla \phi_i \cdot \nabla \phi_j \, ds + \\
& \frac{1}{2} \sum_{i \in I} \sum_{j \in I} v_i v_j \int_{\mathbf{M}} \nabla \phi_i \cdot \nabla \phi_j \, ds + \sum_{i \in I} \sum_{j \in J} v_i v_D(\mathbf{x}_j) \int_{\mathbf{M}} \nabla \phi_i \cdot \nabla \phi_j \, ds + \\
& \sum_{i \in I} \sum_{j \in J} u_D(\mathbf{x}_i) u_D(\mathbf{x}_j) \int_{\mathbf{M}} \nabla \phi_i \cdot \nabla \phi_j \, ds + \sum_{i \in I} \sum_{j \in J} v_D(\mathbf{x}_i) v_D(\mathbf{x}_j) \int_{\mathbf{M}} \nabla \phi_i \cdot \nabla \phi_j \, ds - \\
& \sum_{i \in I} \sum_{j \in J} u_i v_j \int_{\mathbf{M}} \mathbf{n} \cdot (\nabla \phi_i \times \nabla \phi_j) \, ds - \sum_{i \in I} \sum_{j \in J} u_D(\mathbf{x}_i) v_j \int_{\mathbf{M}} \mathbf{n} \cdot (\nabla \phi_i \times \nabla \phi_j) \, ds - \\
& \sum_{i \in I} \sum_{j \in J} u_i v_D(\mathbf{x}_i) \int_{\mathbf{M}} \mathbf{n} \cdot (\nabla \phi_i \times \nabla \phi_j) \, ds - \sum_{i \in I} \sum_{j \in J} u_D(\mathbf{x}_i) v_D(\mathbf{x}_i) \int_{\mathbf{M}} \mathbf{n} \cdot (\nabla \phi_i \times \nabla \phi_j) \, ds
\end{aligned}$$

In order to minimize  $J$ , we can simply cancel the derivative of  $J$  with respect to  $u_k$

$$\begin{aligned}
\frac{\partial J}{\partial u_k} = & \sum_{j \in I} u_j \underbrace{\int_{\mathbf{M}} \nabla \phi_k \cdot \nabla \phi_j \, ds}_{A_{kj}} + \sum_{j \in J} u_D(\mathbf{x}_j) \underbrace{\int_{\mathbf{M}} \nabla \phi_k \cdot \nabla \phi_j \, ds}_{A_{kj}} - \\
& \sum_{j \in I} v_j \underbrace{\int_{\mathbf{M}} \mathbf{n} \cdot (\nabla \phi_k \times \nabla \phi_j) \, ds}_{C_{kj}} - \sum_{j \in I} v_D(\mathbf{x}_j) \underbrace{\int_{\mathbf{M}} \mathbf{n} \cdot (\nabla \phi_k \times \nabla \phi_j) \, ds}_{C_{kj}} \\
= & 0, \quad \forall k \in I.
\end{aligned} \tag{10}$$

The same can be done for the derivative with respect to  $v_k$ .

There are as many equations (10) as there are nodes in  $I$ . This system of equations can be written as:

$$\begin{pmatrix} \bar{\bar{A}} & \bar{\bar{C}} \\ \bar{\bar{C}}^T & \bar{\bar{A}} \end{pmatrix} \begin{pmatrix} \bar{U} \\ \bar{V} \end{pmatrix} = \begin{pmatrix} \bar{0} \\ \bar{0} \end{pmatrix} \tag{11}$$

where  $\bar{\bar{A}}$  is a symmetric positive definite matrix and  $\bar{\bar{C}}$  is an antisymmetric matrix that are both build by assembling the elementary matrices  $A_{kj}$  and  $C_{kj}$ . Hence the resulting matrix in Eq. 11 is symmetric definite positive and efficient direct sparse symmetric-positive-definite solvers such as TAUUS can be used. The vectors  $\bar{U}$  and  $\bar{V}$  denote respectively the vector of unknowns  $u_k$  and  $v_k$ .

In the case of simple Laplacian harmonic maps the matrix  $C$  vanishes, which makes the system of equations (11) uncoupled:

$$\bar{\bar{A}}\bar{\bar{U}} = \bar{0}, \quad \bar{\bar{A}}\bar{\bar{V}} = \bar{0}. \quad (12)$$

Finally, in the case of a convex combination map the system of equations is also uncoupled as in (12), the matrix  $A$  being now the assembly of the following elementary matrices:

$$A_{ij} = \begin{pmatrix} 1 & -0.5 & -0.5 \\ -0.5 & 1 & -0.5 \\ -0.5 & -0.5 & 1 \end{pmatrix} \quad (13)$$

## 6 Boundary conditions

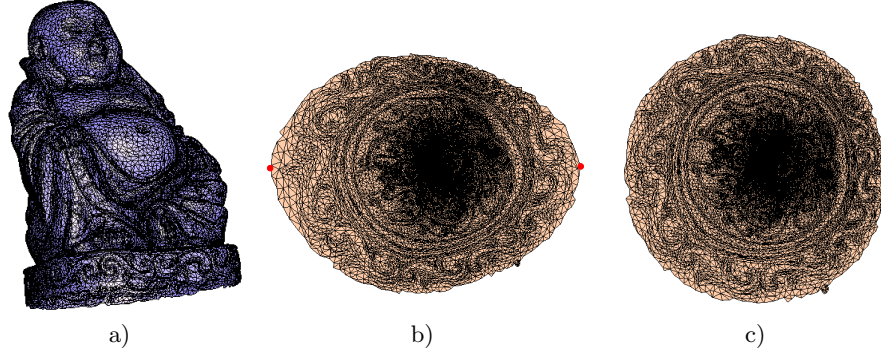
It is necessary to impose appropriate boundary conditions to guarantee that the discrete minimization problem has a unique solution and that this unique solution defines a one-to-one mapping (and hence avoids the degenerate solution  $\mathbf{u} = \text{constant}$ ). Dirichlet boundary conditions are often used for the Laplacian harmonic map and the convex combination map to map the boundary nodes of  $\partial\mathbf{M}_1$  to a unit circle:

$$u_D(\mathbf{x}_i) = \cos\left(\frac{2\pi l_i(\mathbf{x}_i)}{L}\right), \quad v_D(\mathbf{x}_i) = \sin\left(\frac{2\pi l_i(\mathbf{x}_i)}{L}\right). \quad (14)$$

We have decided to map to a unit circle but all kind of convex fixed boundaries could be considered since the mapping is proven to be one-to-one if the mapped surface is convex [29, 7].

Instead of fixing all the boundary nodes  $\partial\mathbf{S}_1$  to a convex polygon, one might fix two  $(u, v)$  coordinates, thus pinning down two vertices in the parameter plane with Dirichlet boundary conditions. Indeed, for least square conformal maps, the mapping (11) has full rank only when the number of pinned vertices is greater or equal to 2 [21]. Pinning down two vertices will set the translation, rotation and scale of the solution when solving the linear system  $\mathbf{L}_C \mathbf{U} = \mathbf{0}$  and will lead to what is called a free-boundary parametrization. It was independendty found by the authors of the LSCM [21] and the DCP [1] that picking two boundary vertices the farthest from each other seems to give good results in general. However, the quality of the conformal parametrization depend drastically on the choice of these constraint vertices. Indeed, global distortion can ensue and a degradation of conformality can be observed around the pinned vertices. Figure 4 compares a LSCM with two pinned vertices with a less distorted LSCM that spreads the constraints throughout the mesh (we call this approach the constrained LSCM or CLSCM)

How can we define a less distorted least square conformal map (CLSCM) without pinning down two vertices ? The idea is to add the two following constraints to the minimization problem that that set the translation, rotation



**Fig. 4.** Initial triangulation  $\mathcal{S}_{\mathcal{T}}$  of a boudda statue a) that has been parametrized by computing b) the LSCM with two constrained vertices (shown in red) c) the constrained LSCM solved with a spectral method.

and scale of the solution: (i) the barycenter of the solution must be at zero and (ii) the moment of inertia of the boundary  $\partial\mathcal{S}_{\mathcal{T}_1}$  must be unit. Those constraints can be taken into account through recourse to spectral theory. This idea was derived also by Mullen et al. [28] and named after **spectral conformal parametrization**. In what follows, we try to present the spectral conformal map in a more comprehensive manner than the way it is presented in [28]. The constrained least square conformal map corresponds to the following discrete constrained minimization problem. Find  $\mathbf{U}^*$  such that

$$\mathbf{U}^* = \arg \min_{\mathbf{U}} \frac{1}{2} \mathbf{U}^T \mathbf{L}_{\mathbf{C}} \mathbf{U}, \quad \text{subject to} \quad \mathbf{U}^T \mathbf{E} = 0, \mathbf{U}^T \mathbf{B} \mathbf{U} = 1. \quad (15)$$

The first constraint in (15)  $\mathbf{U}^T \mathbf{E} = 0$  states that the barycenter of the solution must be at zero. Indeed, as  $\mathbf{E}$  denotes the  $2I \times 2$  matrix that is such that  $E_{i1} = 1, i = 1, \dots, I$  and  $E_{i2} = 1, i = I + 1, \dots, 2I$  (the other entries of  $\mathbf{E}$  being zero), the second constraint  $\mathbf{U}^T \mathbf{B} \mathbf{U} = 1$  indicates that the moment of inertia of the boundary must be unit, the  $\mathbf{B}$  matrix being a  $2I \times 2I$  diagonal matrix with 1 at each diagonal element corresponding to boundary vertices and 0 everywhere else. There are two different ways to solve this constrained minimization problem. The first method tries to find the optimum of the following Lagrangian function  $\mathcal{L}(\mathbf{U}, \boldsymbol{\mu})$  with Lagrange multipliers  $\mu_i \geq 0$ :

$$\mathcal{L}(\mathbf{U}, \lambda \boldsymbol{\mu}) = \frac{1}{2} \mathbf{U}^T \mathbf{L}_{\mathbf{C}} \mathbf{U} - \mu(\mathbf{U}^T \mathbf{E}) - \lambda(\mathbf{U}^T \mathbf{B} \mathbf{U} - 1). \quad (16)$$

The second method is based on spectral theory that shows that the solution of the constrained minimization problem Eq. (15) is the generalized eigenvector  $\mathbf{U}^*$  associated to the smallest non-zero eigenvalue of the matrix  $\mathbf{L}_{\mathbf{C}}$ , i.e the vector satisfying

$$\mathbf{L}_{\mathbf{C}} \mathbf{U} = \lambda \mathbf{B} \mathbf{U}, \quad (17)$$



where  $\lambda$  is the smallest non-zero eigenvalue of  $\mathbf{L}_C$ . This generalized eigenvector  $\mathbf{U}^*$  is called the Fiedler vector of  $\mathbf{L}_C$ . It can be shown that optimizing (16) is equivalent to finding the Fiedler vector  $\mathbf{U}^*$  (17). From a numerical point of view, there exists very efficient eigensolvers that find the Fiedler vector  $\mathbf{U}^*$  of the sparse generalized eigenvalue problem we need to solve (17). Those methods usually proceed through Choleski decomposition (to turn the problem into a conventional eigenvalue problem) and Lanczos iterations, particularly fast in our case since we deal with sparse matrices. Software libraries such as SLEPc [16] or ARPACK [20] provide all those methods for solving the generalized eigenproblem efficiently.

## 7 Results

In this section, we present several remeshing examples in order to compare the three different mapping techniques. We compare timings as well as mesh qualities for the new triangulations or quadrilateral meshes. The quality of the isotropic meshes is evaluated by computing the aspect ratio of every mesh triangle as follows [14]:

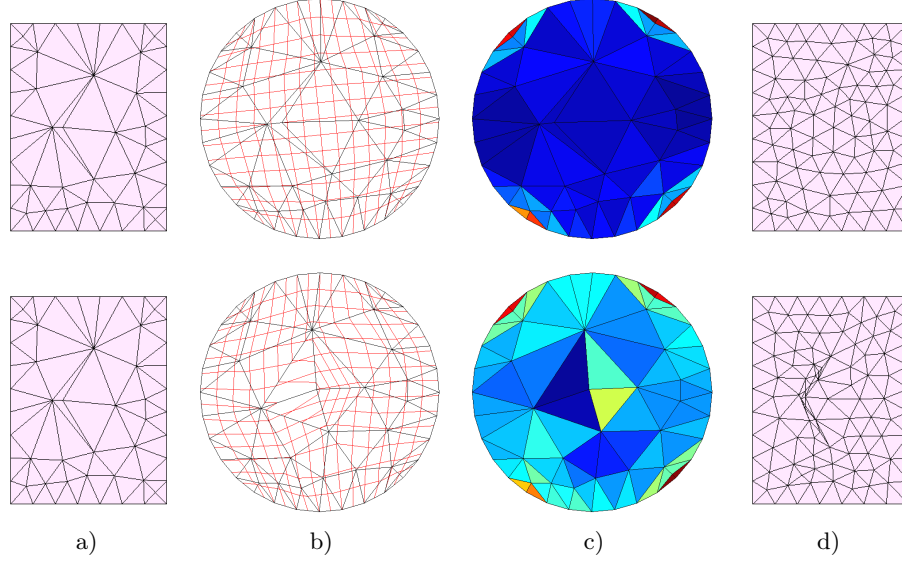
$$\kappa = \alpha \frac{\text{inscribed radius}}{\text{circumscribed radius}} = 4 \frac{\sin \hat{a} \sin \hat{b} \sin \hat{c}}{\sin \hat{a} + \sin \hat{b} + \sin \hat{c}}, \quad (18)$$

$\hat{a}, \hat{b}, \hat{c}$  being the three inner angles of the triangle. With this definition, the equilateral triangle has  $\kappa = 1$  and a flat degenerated triangle has  $\kappa = 0$ . The quality of the quadrangular meshes are evaluated by computing the quality  $\eta$  of every quadrangle as follows:

$$\eta = \max \left( 1 - \frac{2}{\pi} \max_k \left( \left| \frac{\pi}{2} - \alpha_k \right| \right), 0 \right), \quad (19)$$

where  $\alpha_k, k = 1, \dots, 4$  are the four angles of the quadrilateral. This quality measure is  $\eta = 1$  if the element is a perfect quadrilateral and is  $\eta = 0$  if one of those angles is either  $\leq 0$  or  $\geq \pi$ .

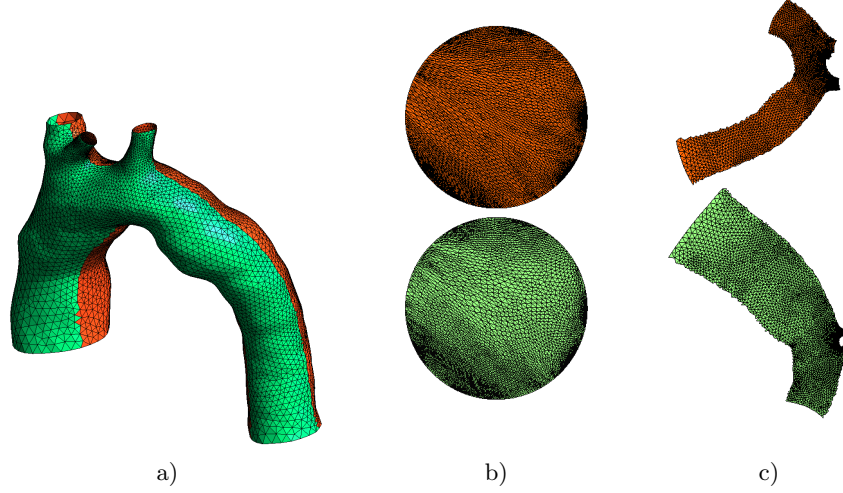
In the first example, we compare the convex combination map and the harmonic map. The convex combination seems attractive from a mathematical point of view and is widely used in the computer graphics community. However, we show in Fig. 5 why in the context of surface remeshing, this mapping should not be used as default mapping. Indeed, as the metric tensor  $\mathbf{M}_u = (\mathbf{x}_u)^T \mathbf{x}_u$  associated with this mapping  $\mathbf{u}(\mathbf{x})$  is much more distorted than the one obtained with the harmonic mapping, there is a negative impact on the quality of the resulting mesh. This is illustrated in Fig. 5 where an initially low quality triangulation  $\mathcal{S}_T$  has been remeshed with a parametrization computed with either the convex combination map or the Laplacian harmonic map. For this example, the total time for the parametrization and the remeshing is 0.008s. A Delaunay planar mesher has been taken for the remeshing in the parametric space.



**Fig. 5.** Poor quality initial triangulation  $\mathcal{S}_{\mathcal{T}}$  (a) that has been remeshed using a harmonic map (top figures) and a convex combination map (bottom figures): b) mapping of the initial mesh onto the unit disk  $\mathcal{S}_{\mathcal{T}'}$  with iso-x and iso-v values in red, c) the determinant of the mesh metric tensor  $\mathbf{M}_{\mathbf{u}}$  that defines the area distortion map and d) the final mesh obtained using the 2D Delaunay algorithms.

In the next example, we compare the remeshing of a human aorta with both the Laplacian and the conformal map. As the geometrical aspect ratio of the triangulation is high, the initial mesh has been automatically split by our algorithm into two different mesh patches. The splitting has been performed with our max-cut mesh partitioner based on a multiscale Laplacian map [24]. As can be seen from Fig. 6, the mapped meshes computed with the Laplacian map present much more distortion close to the boundaries. Again, as most of the planar meshers are more efficient with less distorted meshes, we have that the qualities of the resulting meshes are higher for the conformal map. Indeed, for a radius dependent isotropic remeshing of the aorta, we obtain a minimum quality of  $\kappa_{min} = 0.04$  for the harmonic map and  $\kappa_{min} = 0.39$  for the conformal map. The mean quality is  $\bar{\kappa} = 0.91$  for the harmonic map and  $\bar{\kappa} = 0.96$  for the conformal map. Here, a Frontal planar mesher was chosen for the remeshing in the parametric space. The initial triangulation of the aorta contains 12000 triangles and the remeshing procedure for a new mesh of 5500 triangles took us less than 4s.

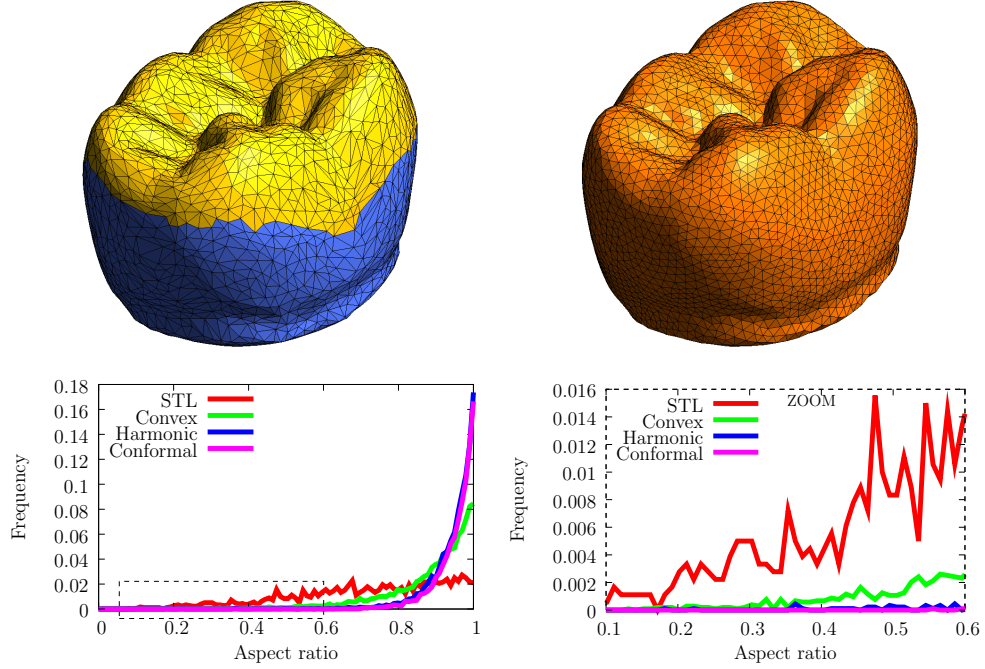
We now compare our three mapping techniques for the remeshing of a tooth of very low quality. Fig. 7 shows the remeshing of the tooth and compares the quality of the remeshing procedure using successively a Laplacian map, a conformal map and a convex combination map and choosing a Frontal



**Fig. 6.** Remeshing of an STL triangulation of a human aorta that has been split into two mesh patches (a). (b) Harmonic mapping and conformal mapping (c) for those two patches.

mesher for the remeshing procedure. As can be seen from Fig.7, the conformal parametrization gives rise to the highest quality mesh while the worst is found to be the convex combination map. The Laplacian mapping has a slightly lower quality that can be explained by a loss in conformality at the boundaries that gives rise to a less smoother mesh metric. The initial triangulation contains 1800 triangles and the remeshed tooth contains about 9000 triangles. The total time for the remeshing is less than 0.8s for all of the three mappings. The specific time for the computation of the convex map is 0.13s, the harmonic map is 0.14s and the conformal map is 0.19s.

An important element in the surface remeshing algorithm is the choice of the planar mesh generator to remesh the parametrized surface. In table 1, we compare the quality of the tooth surface meshes using three different planar mesh generators implemented in Gmsh: a Frontal-Delaunay algorithm [30], a planar Delaunay algorithm [13] and an algorithm based on local mesh adaptation (called MeshAdapt, see [14] for more details). Table 1 shows clearly that the best planar mesh generator is the Gmsh's Frontal-Delaunay algorithm. This is not a surprise: frontal techniques tend to produce meshes that are aligned with principal directions. If the planar domain that has to be meshed is equipped with a smooth metric that conserves angles (i.e., when the mapping is conformal), then the angle between the principal directions is conserved. Frontal algorithms also tend to produce excellent meshes for harmonic maps since harmonic maps are almost conformal except close to the boundaries. The use of Frontal meshers enables us to obtain higher quality mesher from the conformal or harmonic maps. It should be noted that in



**Fig. 7.** Remeshing of a tooth with different mappings and with a planar Delaunay mesh generator. Top figures show the initial STL triangulation and new mesh based on the conformal mapping and the bottom figures show the quality histogram obtained when remeshing the STL file with different mappings. The bottom right figure shows a zoom of the quality histogram.

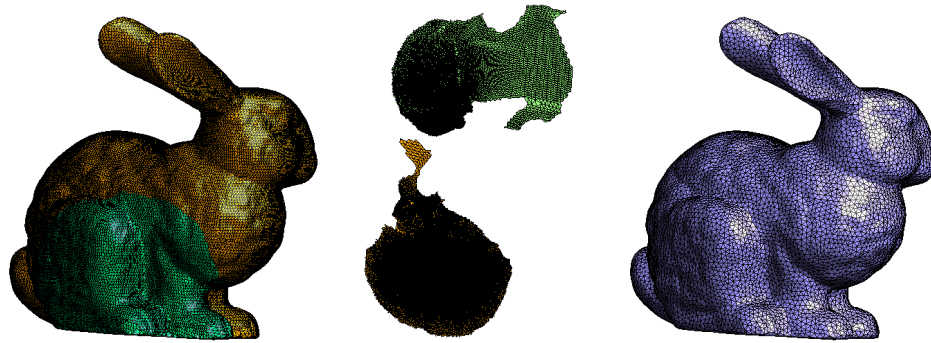
this case the Frontal mesher was not able to build a mesh given the convex combination map.

Mesh generator	Convex map		Harmonic map		Conformal map	
	$\kappa_{min}$	$\bar{\kappa}$	$\kappa_{min}$	$\bar{\kappa}$	$\kappa_{min}$	$\bar{\kappa}$
MeshAdapt	0.05	0.83	0.17	0.94	0.57	0.95
Delaunay	0.002	0.87	0.18	0.94	0.54	0.94
Frontal	-	-	0.36	0.95	0.65	0.96

**Table 1.** Quality of the surface mesh of the tooth using different planar mesh generators for the remeshing of the parametric space, where the parametric space has been computed with the 3 presented mappings. The qualities we look at are the the minimum aspect ratio  $\kappa_{min}$  and the mean aspect ratio  $\bar{\kappa}$ .

Next, we compare the proposed method with other remeshing packages presented in the literature. We consider the well-known stanford bunny mesh

model of 70k triangles<sup>3</sup> (see Fig.8). The original mesh has 5 holes and is of zero genus. For the remeshed bunny of 25k triangles presented in Fig.8, we have a minimum quality of  $\kappa_{min} = 0.56$  and a mean quality is  $\bar{\kappa} = 0.97$ . We compare the timings for the conformal map. Prior to computing the parametrization, two different mesh partitioners have been called: a multilevel mesh partitioner (Metis) and a max-cut mesh partitioner based on a multiscale Laplacian map [24]. The timings for the harmonic and convex map are almost similar. We compare in Table 2 some statistics and timings of our algorithm with the least square conformal map (LSCM) of Levy et al. [21], with the multiresolution remeshing of Eck et al. [8] and with the angle based parametrization (ABF) of Zayer [39]. We can see from Table 2 that our method is quite competitive in terms of computational time with the other methods presented in the literature.



**Fig. 8.** Remeshing of the bunny mesh model of 70k triangles that has been split into 2 mesh partitions. Left figure shows the two partitions, middle figure shows the conformal harmonic parametrization that has been computed for both mesh partitions and right figure shows the remeshed bunny with about 25k triangles.

In the last example, we show that one advantage of conformal mappings is that they can be used to produce quadrilateral meshes given an efficient planar quad meshing algorithm. Indeed, as quadrangular meshes are by definition not isotropic but aligned in some directions, one should have a parametrization that preserve the angles between those directions. Indeed, with such a parametrization, the quads that are created by the planar quad mesh generator will preserve their angles in the 3D space. The only parametrization that preserve angles (in a least square sense) is the conformal parametrization. Figure 9 shows the quadrangular remeshing of half a Falcon aircraft. An initial triangular surface mesh has been generated using a standard frontal surface mesher. Triangles of the surfaces have been patched together to create 12 *com-*

<sup>3</sup>The model can be downloaded at the following web site:  
<http://www.sonycsl.co.jp/person/nielsen/visualcomputing/programs/bunny-conformal.obj>

Remeshing	Number of Partition partitions	Parametrization time (s)	Total remeshing time (s)
LSCM Levy [21] (1.3Ghz)	23	30	95
Eck [8] (1.3Ghz)	88	-	33.5
ABF++ Zayer [39]	2	-	13
LinABF Zayer [39]	2	-	2
Presented method (2.4Ghz)			
* laplacian partitioner	2	16.7	1.4
* multilevel partitioner	10	7	1.4

**Table 2.** Remeshing statistics and timings for the remeshing of the bunny mesh model of 70k triangles (new mesh of 25k triangles) with a conformal map. Comparison (when available) of the presented method with other techniques presented in the Computer Graphics community.

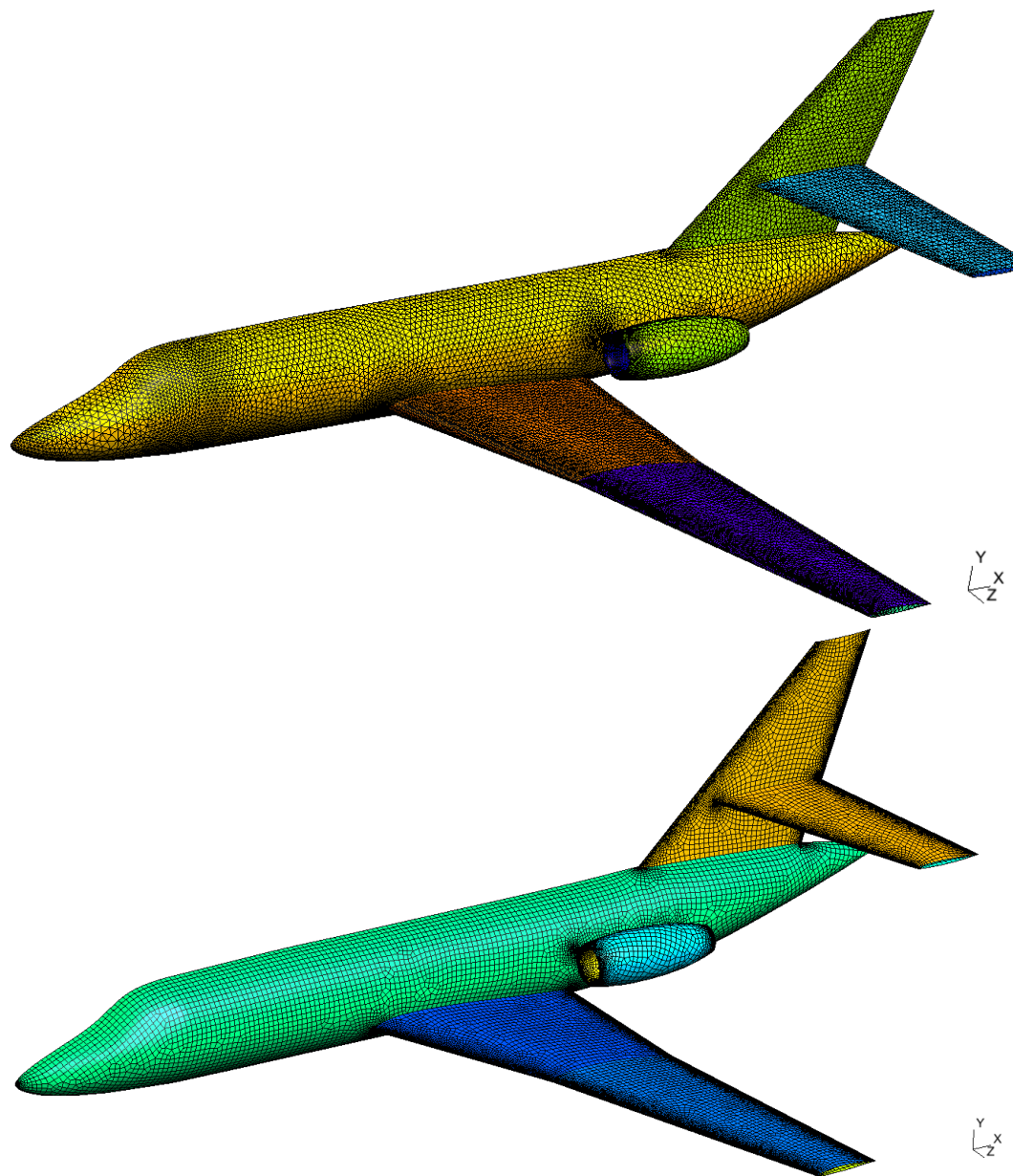
*pounds of surfaces* to be parametrized (only 7 of them are visible in Fig. 9). The colors of the triangles indicate the different compounds of surfaces of the model. The images of the different surfaces on their parameter plane can be seen on Figure 10. The quadrilateral mesh has been obtained with the new Delquad algorithm [31] that generates nearly right triangles combined with the Blossom-quad recombination algorithm [32] that recombines all triangles into quads. The resulting mesh is presented on Figure 9 (bottom). The mesh is composed of 53297 quadrangles. The total time for the surface meshing is 22 seconds. This includes

- The reparametrization of the 12 surfaces (3 sec.),
- The Delquad algorithm applied to the 12 surfaces (10 sec.),
- The Blossom-quad recombination algorithm applied to the 12 surfaces (9 sec.).

For the example of the quad mesh of the Falcon aircraft, the worst and average quality of the mesh are  $\eta_{min} = 0.17$  and  $\bar{\eta} = 0.86$  which can be considered as excellent.

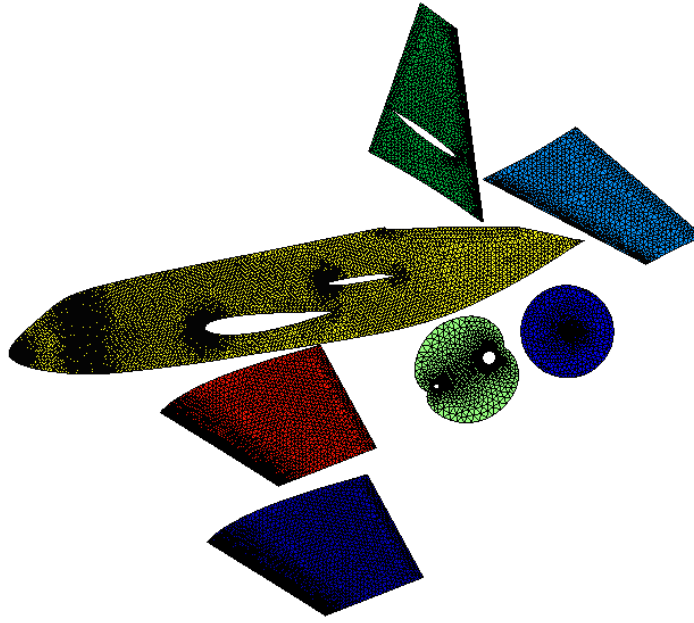
## 8 Conclusion

We presented three different ways to compute discrete mappings. The implementation of those mappings as well as the boundary conditions are presented in a very comprehensive manner such that it becomes accessible to a wider community than the one of computer graphics. We showed that the conformal mapping is the best input for our planar meshes and that this mapping allows to generate high quality meshes both triangular and quadrilateral. The overall remeshing technique based on discrete linear parametrization is efficient and renders high quality meshes.



**Fig. 9.** Initial triangular mesh of half of the Falcon aircraft that has been split into 12 different colored mesh patches (only 7 of those patches are visible) (top) and final quadrangular mesh (bottom).





**Fig. 10.** Spectral conformal parametrization of the 7 visible surfaces patches (see the colored patches on top Fig. 9) of the Falcon aircraft in the parametric plane.

## References

1. P. Alliez, M. Meyer, and M. Desbrun. Interactive geometry remeshing. *Computer graphics (Proceedings of the SIGGRAPH 02)*, pages 347–354, 2002.
2. M. Batdorf, L.A. Freitag, and C. Ollivier-Gooch. A computational study of the effect of unstructured mesh quality on solution efficiency. In *Proc. 13th AIAA Computational Fluid Dynamics Conf.*, 1997.
3. E. Bechet, J.-C. Cuilliere, and F. Trochu. Generation of a finite element mesh from stereolithography (stl) files. *Computer-Aided Design*, 34(1):1–17, 2002.
4. M. Ben-Chen, C. Gotsman, and G. Bunin. Conformal flattening by curvature prescription and metric scaling. *Computer Graphics Forum*, 27(2), 2008.
5. C. Bennis, J.-M. Vézien, and G. Iglésias. Piecewise surface flattening for non-distorted texture mapping. *ACM SIGGRAPH Computer Graphics*, pages 237 – 246, 1991.
6. H Borouchaki, P. Laug, and P.L. George. Parametric surface meshing using a combined advancing-front generalized delaunay approach. *International Journal for Numerical Methods in Engineering*, 49:223–259, 2000.
7. C. Choquet. Sur un type de représentation analytique généralisant la représentation conforme et définie au moyen de fonctions harmoniques. *Bull. Sci. Math*, 69(156-165), 1945.
8. Matthias Eck, Tony DeRose, Tom Duchamp, Hugues Hoppe, Michael Lounsbery, and Werner Stuetzle. Multiresolution analysis of arbitrary meshes. In *SIG-*



- GRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 173–182, 1995.
9. M. S. Floater. Parametrization and smooth approximation of surface triangulations. *Computer aided geometric design*, 14(231-250), 1997.
  10. M. S. Floater. Parametric tilings and scattered data approximation. *International Journal of Shape Modeling*, 4:165–182, 1998.
  11. M. S. Floater. One-to-one piecewise linear mappings over triangulations. *Math. Comp*, 72(685-696), 2003.
  12. M. S. Floater and K. Hormann. Surface parameterization: a tutorial and survey. *Advances in Multiresolution for Geometric Modelling*, 2005.
  13. P.-L. George and P. Frey. *Mesh Generation*. Hermes, 2000.
  14. C. Geuzaine and J.-F. Remacle. Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 79(11):1309–1331, 2009.
  15. G. Greiner and K. Hormann. Interpolating and approximating scattered 3d data with hierarchical tensor product splines. In *Surface Fitting and Multiresolution Methods*, pages 163–172, 1996.
  16. V. Hernandez, J.E. Roman, and V. Vidal. Slepnc: A scalable and flexible toolkit for the solution of eigenvalue problems. *ACM Trans. Math. Soft.*, 31(3):351–362, 2005.
  17. K. HORMANN, , and G. 2000 GREINER. Mips: An efficient global parametrization method. In *Curve and Surface Design*. Vanderbilt University Press, 2000.
  18. Y. Ito and K Nakahashi. Direct surface triangulation using stereolithography data. *AIAA Journal*, 40(3):490–496, 2002.
  19. P. Laug and H. Boruchaki. Interpolating and meshing 3d surface grids. *International Journal for Numerical Methods in Engineering*, 58:209–225, 2003.
  20. R. B. Lehoucq, D. C. Sorensen, and C. Yang. *ARPACK Users Guide: Solution of Large Scale Eigenvalue Problems by Implicitly Restarted Arnoldi Methods*. Society for Industrial and Applied Mathematics, 1997.
  21. B. Levy, S. Petitjean, N. Ray, and J. Maillot. Least squares conformal maps for automatic texture atlas generation. In *Computer Graphics (Proceedings of SIGGRAPH 02)*, pages 362 – 371, 2002.
  22. M. Spagnuolo M. Attene, B. Falcidieno and G. Wyvill. A mapping-independent primitive for the triangulation of parametric surfaces. *Graphical Models*, 65(260-273), 2003.
  23. J. Maillot, H. Yahia, and A. Verroust. Interactive texture mapping. In *Proceedings of ACM SIGGRAPH'93*, pages 27–34, 1993.
  24. E. Marchandise, C. Carton de Wiart, W.G. Vos, C. Geuzaine, and J-F. Remacle. High quality surface remeshing using harmonic maps. Part II: Surfaces with high genus and of large aspect ratio. *International Journal for Numerical Methods in Engineering*, 86(11):1303–1321, June 2011.
  25. E. Marchandise, G. Compère, M. Willemet, G. Bricteux, C. Geuzaine, and J.-F. Remacle. Quality meshing based on stl triangulations for biomedical simulations. *International Journal for Numerical Methods in Biomedical Engineering*, 83:876–889, 2010.
  26. David L. Marcum. Efficient generation of high-quality unstructured surface and volume grids. *Engrg. Comput.*, 17:211–233, 2001.

27. David L. Marcum and Adam Gaither. Unstructured surface grid generation using global mapping and physical space approximation. In *Proceedings, 8th International Meshing Roundtable*, pages 397–406, 1999.
28. P. Mullen, Y. Tong, P. Alliez, and M. Desbrun. Spectral conformal parameterization. In *ACM/EG Symposium of Geometry Processing*, 2008.
29. T. Rado. Aufgabe 41. *Math-Verien*, page 49, 1926.
30. S. Rebay. Efficient unstructured mesh generation by means of delaunay triangulation and bowyer-watson algorithm. *Journal of Computational Physics*, 106:25–138, 1993.
31. J-F. Remacle, F. Henrotte, T. Carrier-Baudouin, E. Bechet, E. Marchandise, C. Geuzaine, and T. Mouton. A frontal delaunay quad mesh generator using the  $l_\infty$  norm. *International Journal for Numerical Methods in Engineering*, 2011.
32. J.-F. Remacle, J. Lambrechts, B. Seny, E. Marchandise, A. Johnen, and C. Geuzaine. Blossom-quad: a non-uniform quadrilateral mesh generator using a minimum cost perfect matching algorithm. *International Journal for Numerical Methods in Engineering*, 2011. submitted.
33. A. Sheffer, E. Praun, and K. Rose. Mesh parameterization methods and their applications. *Found. Trends. Comput. Graph. Vis.*, 2(2):105–171, 2006.
34. Alla Sheffer, Bruno Lévy, Inria Lorraine, Maxim Mogilnitsky, and Er Bogomyakov. Abf++: fast and robust angle based flattening. *ACM Transactions on Graphics*, 24(311-330), 2005.
35. D. Szczerba, R. McGregor, and G. Szekely. High quality surface mesh generation for multi-physics bio-medical simulations. In *Computational Science – ICCS 2007*, volume 4487, pages 906–913. Springer Berlin, 2007.
36. J.R. Tristano, S.J. Owen, and S.A. Canann. Advancing front surface mesh generation in parametric space using riemannian surface definition. In *Proceedings of 7th International Meshing Roundtable*, pages 429–455. Sandia National Laboratory, 1998.
37. W.T Tutte. How to draw a graph. In *Proceedings of the London Mathematical Society*, volume 13, pages 743–768, 1963.
38. D. Wang, O. Hassan, K. Morgan, and N. Weatheril. Enhanced remeshing from stl files with applications to surface grid generation. *Commun. Numer. Meth. Engng*, 23:227–239, 2007.
39. Rhaleb Zayer, Bruno Lévy, and Hans-Peter Seidel. Linear angle based parameterization. In *ACM/EG Symposium on Geometry Processing conference proceedings*, 2007.
40. Y. Zheng, N.P. Weatherill, and O. Hassan. Topology abstraction of surface models for three-dimensional grid generation. *Engrg. Comput.*, 17(28-38), 2001.
41. Gil Zigelman, Ron Kimmel, and Nahum Kiryati. Texture mapping using surface flattening via multi-dimensional scaling. *IEEE Trans. on Visualisation and Computer Graphics*, 8(198-207), 2002.